

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Facultad de Ingeniería
Escuela de Ciencias y Sistemas



Proyecto 2

Grupo 6

Sistemas de Bases de datos 2

Luis Antonio Castillo Javier	202003745
Brian Estuardo Ajuchán Tocoeh	202001086
Rosa Emilia Boche Naz	200915633

Modelo de datos utilizado en MongoDB

Para la creación de la base de datos NoSQL utilizando MongoDB, se desarrolló un esquema flexible basado en documentos, que facilita el almacenamiento eficiente de información clínica. A diferencia de los modelos de las bases de datos relacionales (como MySQL), MongoDB permite el uso de documentos anidados y colecciones con estructuras dinámicas, lo que ofrece una mejor adaptación a la naturaleza variable de los datos.

Colecciones definidas

Se diseñaron las siguientes colecciones para representar los archivos proporcionados:

PACIENTE

Contiene la información general de cada paciente.

HABITACIONES

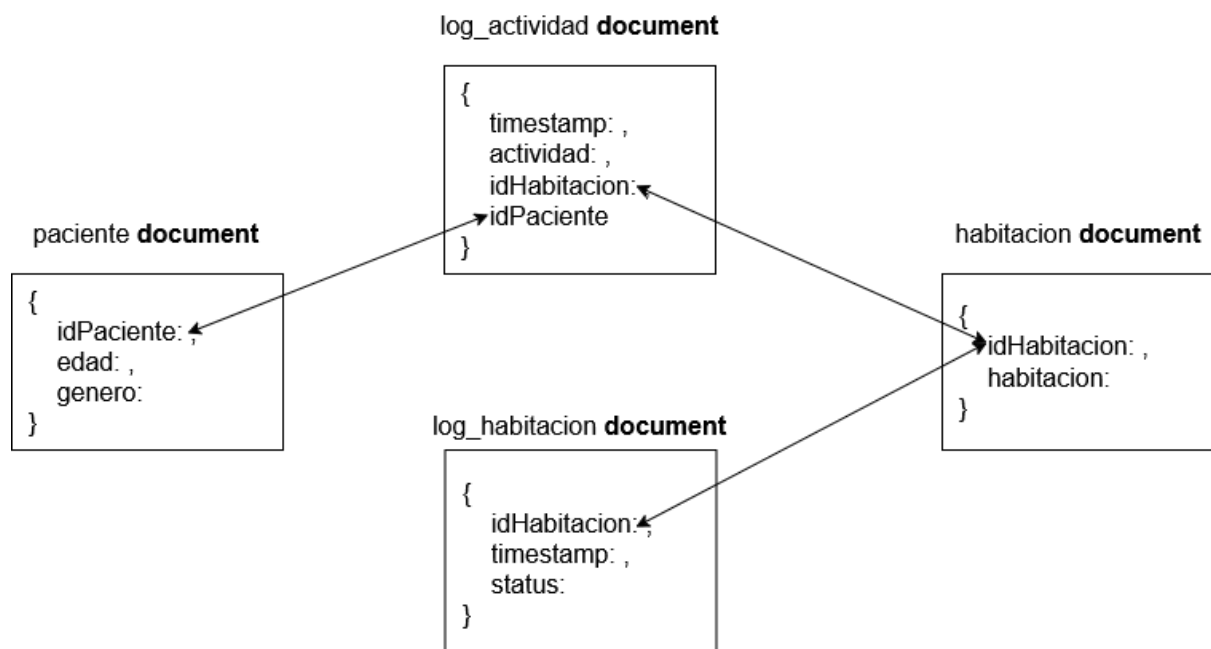
Información sobre cada habitación disponible en la clínica

LOG_HABITACIONES

Registro de cada vez que un paciente ha ocupado una habitación.

LOG_ACTIVIDADES

Registro de todas las actividades realizadas por los pacientes dentro de la clínica.



Modelo utilizado mysql

Para la implementación del modelo relacional, se utilizó el dbms MYSQL, creando diferentes entidades y relaciones para la administración de pacientes y habitaciones, además de sus historiales.

Las entidades que se implementan son:

PACIENTE

ATRIBUTOS

- id_paciente
- edad
- genero

HABITACION

ATRIBUTOS

- id_habitacion
- habitacion

LOG_HABITACION

ATRIBUTO

- id_log_habitacion
- timestamp
- status
- id_habitacion

LOG_ACTIVIDAD

ATRIBUTOS

- id_log_actividad
- timestamp
- actividad
- idHabitacion
- idPaciente

LOS SCRIP PARA GENERAR LA BASE DE DATOS

```
CREATE DATABASE Clinica;
```

```
USE Clinica;
```

```
-- Tabla Paciente
```

```
CREATE TABLE PACIENTE (  
    idPaciente INT PRIMARY KEY,  
    edad INT NOT NULL,  
    genero VARCHAR(20) NOT NULL  
);
```

```
-- Tabla Habitacion
```

```
CREATE TABLE HABITACION (  
    idHabitacion INT PRIMARY KEY,  
    habitacion VARCHAR(50) NOT NULL  
);
```

```
-- Tabla Log de Actividad
```

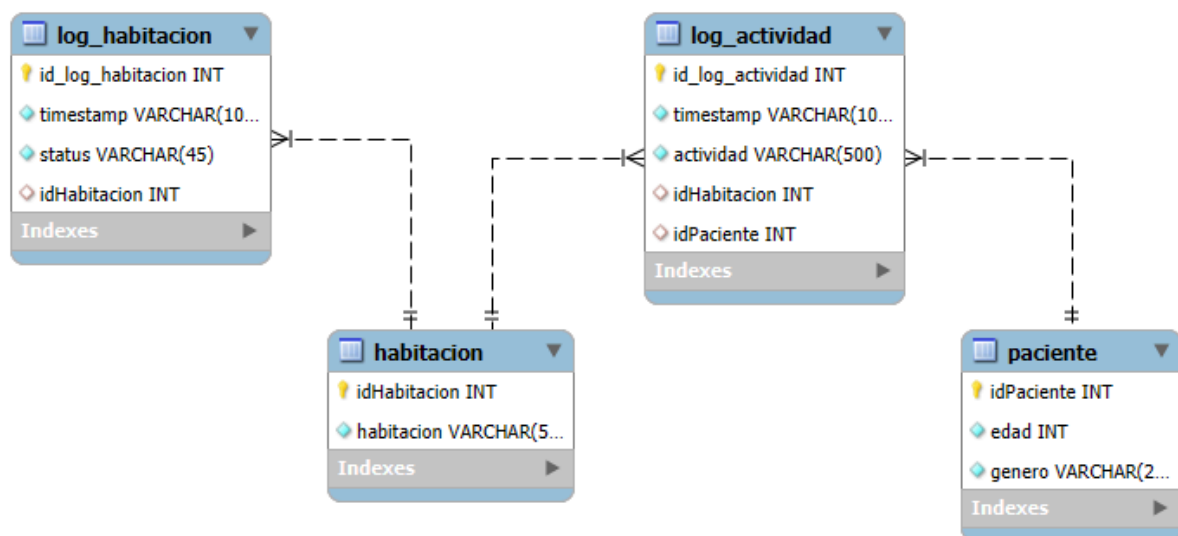
```
CREATE TABLE LOG_ACTIVIDAD (  
    id_log_actividad INT PRIMARY KEY,  
    timestamp VARCHAR(100) NOT NULL,  
    actividad VARCHAR(500) NOT NULL,  
    idHabitacion INT,  
    idPaciente INT,  
    FOREIGN KEY (idPaciente) REFERENCES PACIENTE(idPaciente),  
    FOREIGN KEY (idHabitacion) REFERENCES HABITACION(idHabitacion)
```

);

-- Tabla Log de Habitacion

```
CREATE TABLE LOG_HABITACION (  
    id_log_habitacion INT PRIMARY KEY,  
    timestamp VARCHAR(100) NOT NULL,  
    status VARCHAR(45) NOT NULL,  
    idHabitacion INT,  
    FOREIGN KEY (idHabitacion) REFERENCES HABITACION(idHabitacion)  
);
```

MODELO ENTIDAD RELACION



Respuestas esperadas de la API

Para la realización de este proyecto, se utilizó una api en nodejs, la cual tiene conexión con dos clientes de bases de datos, Mongo y Mysql, a continuación se darán los detalles de las respuestas esperadas en la API

MySQL

Pacientes por categoría:

```
Body Cookies Headers (7) Test Results 200 OK 63 ms 422 B Save Response
{} JSON Preview Visualize
1 {
2   "mensaje": "Distribución por categorías de edad obtenida correctamente.",
3   "categorias": {
4     "pediatrico": "37841",
5     "mediana_edad": "82254",
6     "geriatrico": "34889"
7   },
8   "tiempoRespuestaMs": "57.29 ms"
9 }
```

Pacientes por habitación:

```
Body Cookies Headers (7) Test Results 200 OK 82 ms 1.27 KB Save Response
{} JSON Preview Visualize
53 {
54   "habitacion": "Sala de procedimientos 2",
55   "cantidadPacientes": 1350
56 },
57 {
58   "habitacion": "Sala de procedimientos 3",
59   "cantidadPacientes": 1022
60 },
61 {
62   "habitacion": "Sala de procedimientos 4",
63   "cantidadPacientes": 760
64 },
65 ],
66 "tiempoRespuestaMs": "79.41 ms"
67 }
68 }
```

Pacientes por género

```
Body Cookies Headers (7) Test Results 200 OK 94 ms 481 B Save Response
{} JSON Preview Visualize
1 {
2   "mensaje": "Cantidad de pacientes agrupados por género obtenida correctamente.",
3   "datos": {
4     "datos": [
5       {
6         "genero": "Otro",
7         "cantidad": 5445
8       },
9       {
10        "genero": "Femenino",
11        "cantidad": 75214
12      },
13      {
14        "genero": "Masculino",
15        "cantidad": 73525
16      }
17    ],
18    "tiempoRespuestaMs": "88.70 ms"
19  }
20 }
```

Top edades mas atendidas

Body Cookies Headers (7) Test Results 200 OK 57 ms 490 B Save Response

{ } JSON Preview Visualize

```
11      "cantidad": 2418
12    },
13    {
14      "edad": 2,
15      "cantidad": 2391
16    },
17    {
18      "edad": 15,
19      "cantidad": 2389
20    },
21    {
22      "edad": 11,
23      "cantidad": 2382
24    }
25  ],
26  "tiempoRespuestaMs": "52.42 ms"
27 }
28 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Top edades menos atendidas

Body Cookies Headers (7) Test Results 200 OK 50 ms 489 B Save Response

{ } JSON Preview Visualize

```
11      "cantidad": 756
12    },
13    {
14      "edad": 79,
15      "cantidad": 771
16    },
17    {
18      "edad": 77,
19      "cantidad": 777
20    },
21    {
22      "edad": 83,
23      "cantidad": 778
24    }
25  ],
26  "tiempoRespuestaMs": "45.50 ms"
27 }
28 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Top habitaciones más utilizadas

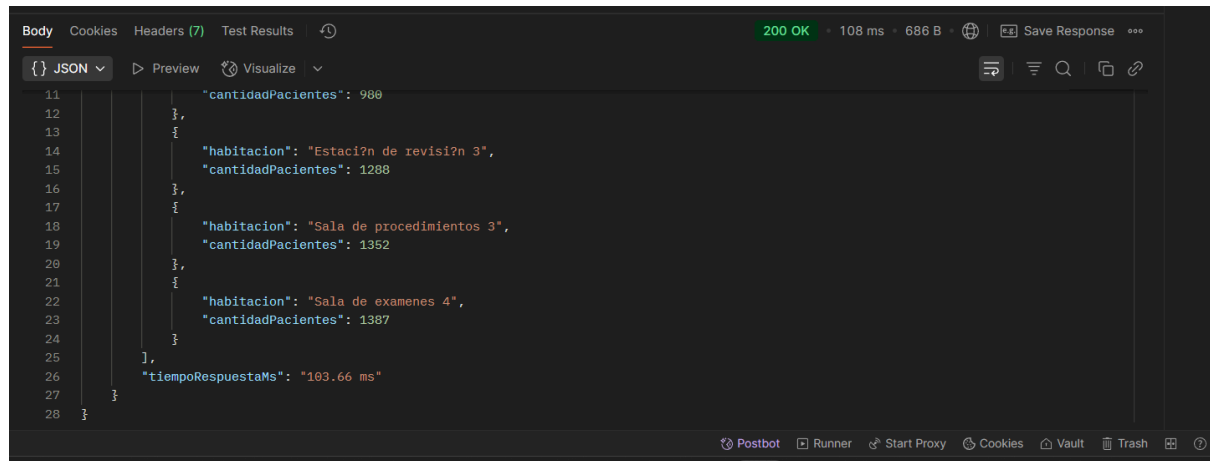
Body Cookies Headers (7) Test Results 200 OK 97 ms 666 B Save Response

{ } JSON Preview Visualize

```
11      "cantidadPacientes": 5286
12    },
13    {
14      "habitacion": "Sala de exámenes 1",
15      "cantidadPacientes": 3778
16    },
17    {
18      "habitacion": "Sala de exámenes 2",
19      "cantidadPacientes": 2968
20    },
21    {
22      "habitacion": "Estaci?n de revisi?n 2",
23      "cantidadPacientes": 2790
24    }
25  ],
26  "tiempoRespuestaMs": "93.95 ms"
27 }
28 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Top habitaciones menos utilizadas



A screenshot of a REST client interface showing a JSON response. The status bar at the top indicates a 200 OK response with a 108 ms response time and 686 B of data. The JSON body is as follows:

```
11 {
12   "cantidadPacientes": 900
13 },
14 {
15   "habitacion": "Estaci?n de revisi?n 3",
16   "cantidadPacientes": 1288
17 },
18 {
19   "habitacion": "Sala de procedimientos 3",
20   "cantidadPacientes": 1352
21 },
22 {
23   "habitacion": "Sala de exámenes 4",
24   "cantidadPacientes": 1387
25 },
26 ],
27 "tiempoRespuestaMs": "103.66 ms"
28 }
```

Día con más pacientes

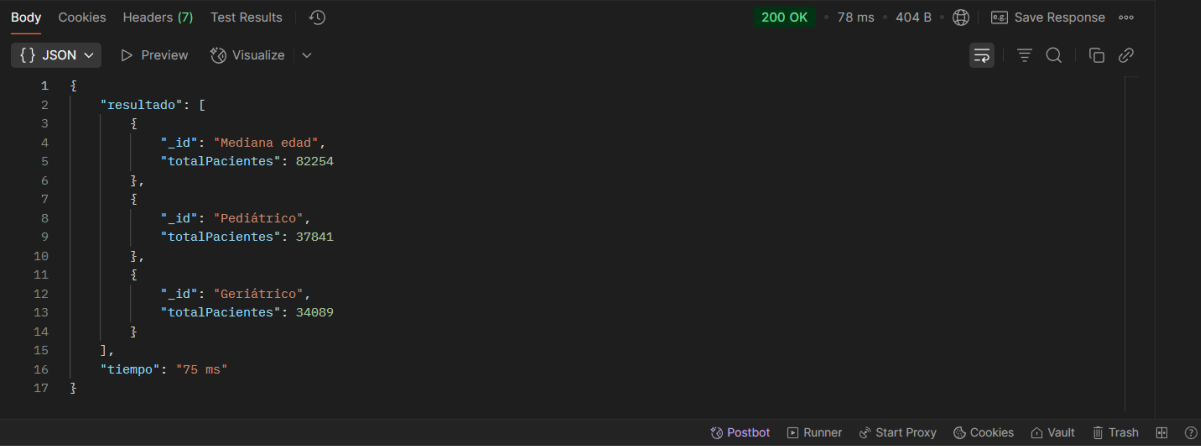


A screenshot of a REST client interface showing a JSON response. The status bar at the top indicates a 200 OK response with a 56 ms response time and 406 B of data. The JSON body is as follows:

```
1 {
2   "mensaje": "Día con más pacientes obtenido correctamente.",
3   "datos": {
4     "datos": {
5       "dia": "2021-07-21T06:00:00.000Z",
6       "cantidadPacientes": 1476
7     },
8     "tiempoRespuestaMs": "50.91 ms"
9   }
10 }
```


MongoDB

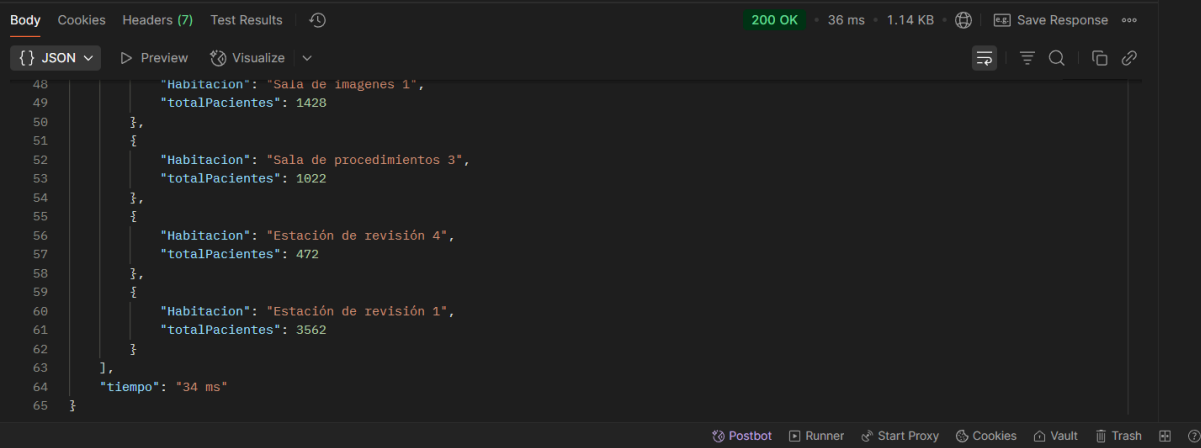
Total pacientes por categoría



The screenshot shows a REST client interface with a 200 OK status, 78 ms response time, and 404 B body size. The JSON response is as follows:

```
1 {
2   "resultado": [
3     {
4       "_id": "Mediana edad",
5       "totalPacientes": 82254
6     },
7     {
8       "_id": "Pediátrico",
9       "totalPacientes": 37841
10    },
11    {
12      "_id": "Geriátrico",
13      "totalPacientes": 34089
14    }
15  ],
16  "tiempo": "75 ms"
17 }
```

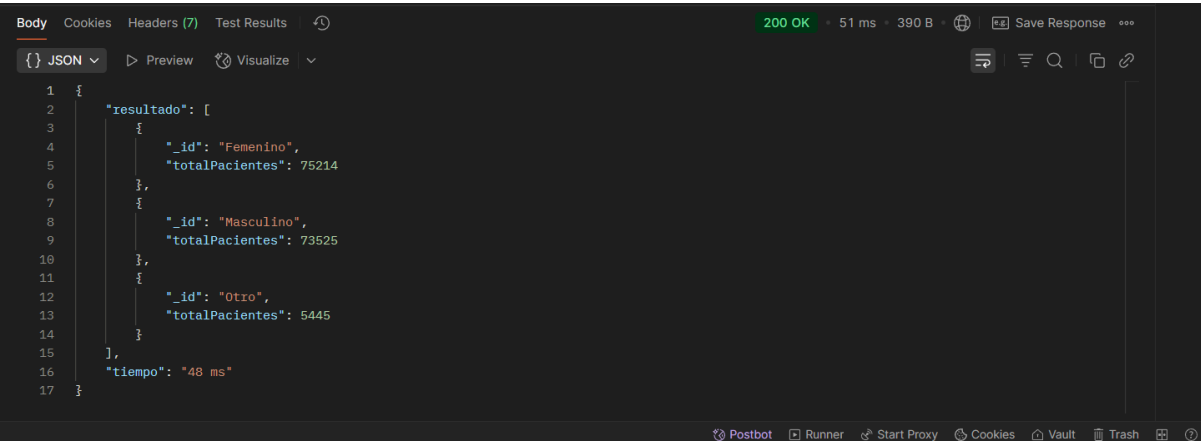
Pacientes por habitación



The screenshot shows a REST client interface with a 200 OK status, 36 ms response time, and 1.14 KB body size. The JSON response is as follows:

```
48   "Habitacion": "Sala de imagenes 1",
49   "totalPacientes": 1428
50 },
51 {
52   "Habitacion": "Sala de procedimientos 3",
53   "totalPacientes": 1022
54 },
55 {
56   "Habitacion": "Estación de revisión 4",
57   "totalPacientes": 472
58 },
59 {
60   "Habitacion": "Estación de revisión 1",
61   "totalPacientes": 3562
62 },
63 ],
64 "tiempo": "34 ms"
65 }
```

Pacientes por género



The screenshot shows a REST client interface with a 200 OK status, 51 ms response time, and 390 B body size. The JSON response is as follows:

```
1 {
2   "resultado": [
3     {
4       "_id": "Femenino",
5       "totalPacientes": 75214
6     },
7     {
8       "_id": "Masculino",
9       "totalPacientes": 73525
10    },
11    {
12      "_id": "Otro",
13      "totalPacientes": 5445
14    }
15  ],
16  "tiempo": "48 ms"
17 }
```

Top edades atendidas

BodyCookiesHeaders (7)Test Results🕒

200 OK · 48 ms · 400 B · 🌐 📄 Save Response ⋮

{ } JSON ▾▶ Preview🔗 Visualize ▾

```
8      {
9        "_id": 8,
10       "cantidad": 2418
11     },
12     {
13       "_id": 2,
14       "cantidad": 2391
15     },
16     {
17       "_id": 15,
18       "cantidad": 2389
19     },
20     {
21       "_id": 11,
22       "cantidad": 2382
23     }
24   ],
25   "tiempo": "44 ms"
}
```

🔗 Postbot 📄 Runner ⚙️ Start Proxy 🍪 Cookies 🏠 Vault 🗑️ Trash 🧩 ⓘ

Top edades menos atendidas

BodyCookiesHeaders (7)Test Results🕒

200 OK · 48 ms · 398 B · 🌐 📄 Save Response ⋮

{ } JSON ▾▶ Preview🔗 Visualize ▾

```
14     {
15       "_id": 77,
16       "cantidad": 777
17     },
18     {
19       "_id": 83,
20       "cantidad": 778
21     }
22   ],
23   "tiempo": "44 ms"
24 }
25 }
```

🔗 Postbot 📄 Runner ⚙️ Start Proxy 🍪 Cookies 🏠 Vault 🗑️ Trash 🧩 ⓘ

Top habitaciones más utilizadas

BodyCookiesHeaders (7)Test Results🕒

200 OK · 27 ms · 563 B · 🌐 📄 Save Response ⋮

{ } JSON ▾▶ Preview🔗 Visualize ▾

```
14     {
15       "Habitacion": "Sala de exámenes 2",
16       "totalPacientes": 2968
17     },
18     {
19       "Habitacion": "Estación de revisión 2",
20       "totalPacientes": 2790
21     }
22   ],
23   "tiempo": "23 ms"
24 }
25 }
```

🔗 Postbot 📄 Runner ⚙️ Start Proxy 🍪 Cookies 🏠 Vault 🗑️ Trash 🧩 ⓘ

Top habitaciones menos usadas

BodyCookiesHeaders (7)Test Results🕒

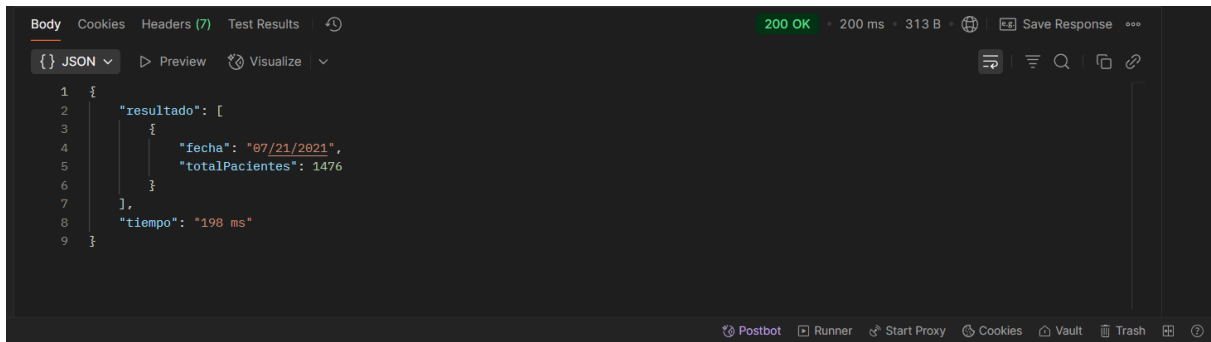
200 OK · 27 ms · 581 B · 🌐 📄 Save Response ⋮

{ } JSON ▾▶ Preview🔗 Visualize ▾

```
14     {
15       "Habitacion": "Sala de procedimientos 3",
16       "totalPacientes": 1352
17     },
18     {
19       "Habitacion": "Sala de exámenes 4",
20       "totalPacientes": 1387
21     }
22   ],
23   "tiempo": "23 ms"
24 }
25 }
```

🔗 Postbot 📄 Runner ⚙️ Start Proxy 🍪 Cookies 🏠 Vault 🗑️ Trash 🧩 ⓘ

Día con menos pacientes



Link para ver mejor la documentación en postman

<https://documenter.getpostman.com/view/34331086/2sB2j68pav>

Conclusión y justificación del trabajo realizado

Luego de realizar la implementación del sistema de datos en ambos gestores MySQL (relacional) y MongoDB (NoSQL) y llevar a cabo las respectivas consultas solicitadas por la clínica, se pudo evidenciar una diferencia significativa en el rendimiento de ambos sistemas.

MongoDB mostró tiempos de respuesta un poco menores en comparación con MySQL, especialmente en operaciones que implican manejo de grandes volúmenes de datos o estructuras anidadas. Esto se debe a su naturaleza orientada a documentos, que permite una recuperación más directa de la información sin necesidad de realizar múltiples JOINS como en el modelo relacional. Además, la flexibilidad del esquema de MongoDB permite adaptarse con mayor facilidad a los cambios y crecimiento de la información clínica sin necesidad de reestructurar toda la base de datos.

Una de las principales razones por las cuales MySQL presentó tiempos de respuesta más altos es porque su modelo relacional obliga a dividir la información en múltiples tablas relacionadas entre sí, lo que requiere realizar operaciones de **JOIN** para obtener resultados compuestos. Estas operaciones pueden volverse costosas en términos de rendimiento cuando se manejan grandes cantidades de datos o relaciones complejas. Por el contrario, MongoDB almacena los datos en documentos auto-contenidos (generalmente en formato BSON), lo cual permite acceder a toda la información relevante sin necesidad de hacer múltiples consultas ni unir tablas. Esta estructura hace que MongoDB sea **más eficiente y rápido** en escenarios como el de la clínica, donde los datos pueden consultarse de forma directa y sin una estructura rígida.

Aunque MySQL es una herramienta robusta, confiable y ampliamente utilizada, su rigidez estructural y los tiempos de procesamiento más largos para ciertas consultas lo hacen menos adecuado para entornos altamente dinámicos como el de una clínica médica moderna.

Se concluye que la adopción de MongoDB como gestor principal de base de datos resulta ser una opción viable y beneficiosa para la clínica, no solo por su mejor desempeño en tiempo de respuesta, sino también por su flexibilidad, escalabilidad y facilidad de integración con sistemas modernos.